A Consistency-Aware Spot-Guided Transformer for Versatile and Hierarchical Point Cloud Registration

Renlang Huang **Yufan Tang** **Jiming Chen** **Liang Li***
College of Control Science and Engineering
Zhejiang University, Hangzhou 310027, China
{renlanghuang,tyfan,cjm,liang.li}@zju.edu.cn

Abstract

Deep learning-based feature matching has shown great superiority for point cloud registration in the absence of pose priors. Although coarse-to-fine matching approaches are prevalent, the coarse matching of existing methods is typically sparse and loose without consideration of geometric consistency, which makes the subsequent fine matching rely on ineffective optimal transport and hypothesis-and-selection methods for consistency. Therefore, these methods are neither efficient nor scalable for real-time applications such as odometry in robotics. To address these issues, we design a consistency-aware spot-guided Transformer (CAST), which incorporates a spot-guided cross-attention module to avoid interfering with irrelevant areas, and a consistency-aware self-attention module to enhance matching capabilities with geometrically consistent correspondences. Furthermore, a lightweight fine matching module for both sparse keypoints and dense features can estimate the transformation accurately. Extensive experiments on both outdoor LiDAR point cloud datasets and indoor RGBD point cloud datasets demonstrate that our method achieves *state-of-the-art* accuracy, efficiency, and robustness. Our code is available at `https://github.com/RenlangHuang/CAST`.

1 Introduction

Point cloud registration is a fundamental yet crucial task for a variety of 3D vision and robotic applications, such as simultaneous localization and mapping (SLAM) [1], object pose estimation [2] and structure from motion (SfM) [3]. Aiming at aligning two partially overlapped point clouds, the typical approach involves a two-stage pipeline: data association which establishes reliable point correspondences, and pose estimation. However, establishing these correspondences has been challenging due to the noisy, irregular, non-uniform, and textureless nature of 3D point clouds.

Feature matching has long been the mainstream of data association without pose priors. Extensive research has made advances in distinctive local feature representations, ranging from hand-crafted descriptors [4, 5, 6] to recent learning-based descriptors [7, 8, 9]. Although the emerging learning-based descriptors significantly improve the reliability of correspondences, the inlier ratio still falls short of what is required for robust and efficient pose estimation. Recently, coarse-to-fine matching is a thriving framework for 2D-2D [10], 3D-3D [11, 12, 13], and even 2D-3D [14] data association. It has been a consensus that Transformers stacked by alternate self-attention and cross-attention modules are effective for coarse matching, which are inspired by human visual processes. Typically, humans may first scan through the point clouds to identify and match salient landmarks across different point clouds reliably. For less salient points, the geometric relationships between them and those salient landmarks would be utilized to revisit their potential correspondences. The correspondences will eventually be established for the entire point cloud after several iterations of this process.

*Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Unfortunately, existing coarse matching approaches tend to be sparse and loose without consideration of geometric consistency. An important reason for looseness is that global cross-attention inevitably attends to similar yet irrelevant areas, resulting in misleading feature aggregation and consequent inconsistent correspondences that undermine both robustness and accuracy. As a result, the hypothesis-and-selection pipeline such as RANSAC [15] is commonly used for outlier rejection, which is typically inaccurate and inefficient, especially for numerous samples with low inlier ratio. Furthermore, the sparsity necessitates the use of complicated fine matching such as optimal transport-based algorithms to establish reliable dense correspondences. Due to iterative dense matrix operations for patch-to-patch correspondences established by coarse matching, these fine matching methods are neither efficient nor scalable for real-time large-scale applications such as odometry.

To this end, we attempt to design an efficient and scalable coarse-to-fine matching network based on consistency-aware semi-dense coarse correspondences. Inspired by ASTR [10] for 2D feature matching, we leverage local consistency to direct the cross-attention of each point exclusively to corresponding patches of its confident neighbors, which is referred to as spot-guided cross-attention. Unlike [10], we propose a novel consistency-aware matching confidence criterion to sample reliable neighbors based on both feature similarity and geometric compatibility. Additionally, we design a consistency-aware self-attention module to enhance the distinctiveness of coarse feature representations via aggregation with salient nodes from the compatibility graph. Notably, both spot-guided cross-attention and consistency-aware self-attention are efficient sparse attention mechanisms.

For scalability to real-time applications such as odometry with pose priors, we propose a lightweight fine matching module allowing independent deployment without coarse matching. The scalability is credited to flexible point-to-patch local matching instead of optimal transport heavily relying on patch-to-patch correspondences. In addition, our fine matching adopts a sparse-to-dense registration pipeline, benefiting from the efficiency of sparse keypoint matching and the accuracy of dense registration. Furthermore, an efficient compatibility graph embedding module is leveraged for outlier rejection as a substitute for inefficient hypothesis-and-selection pipelines.

In summary, our main contributions are as follows:

- A consistency-aware spot-guided Transformer (CAST) with multi-scale feature fusion for much tighter coarse matching with a focus on geometric consistency.
- A spot-guided cross-attention module with a consistency-aware matching confidence criterion that can maintain local consistency without interfering with irrelevant areas.
- A consistency-aware self-attention module based on sparse sampling from the compatibility graph to enhance global consistency during feature aggregation.
- A lightweight and scalable sparse-to-dense matching module involving both sparse keypoints and dense features to achieve lower registration errors without optimal transport and hypothesis-and-selection pipelines.

2 Related Work

3D Feature Descriptors. Feature matching plays a crucial role in point cloud registration, enabling the establishment of reliable correspondences without pose priors. Early methods use hand-crafted descriptors based on signatures [6] or histograms [4, 5] to represent local geometric features. Recently, learning-based 3D descriptors have showcased greater performance than hand-crafted ones, which are usually trained in a self-supervised manner by maximizing the similarity between descriptors of true correspondences and minimizing the similarity otherwise. 3DMatch [16] and PerfectMatch [7] leverage 3D CNNs to learn local patch-wise descriptors from 3D patches converted into voxels of truncated distance function (TDF) values and smoothed density value (SDV) representations, respectively. PPFNet [17] extracts global context-aware patch-wise descriptors based on PointNet [18]. FCGF [8] employs a sparse 3D convolutional encoder-decoder network for dense descriptor learning. SpinNet [19] proposes a 3D cylindrical convolution network to extract rotation-invariant patch-wise descriptors. Predator [20] utilizes graph convolution and cross-attention to enhance the descriptors and predict the overlapping regions for robust performance in low overlap scenarios.

3D Keypoint Detectors. Detection-based methods have been widely studied in image matching but less developed for 3D point clouds. Existing 3D keypoint detectors are mainly hand-crafted, which

extract salient points based on unique geometric features such as specific curvatures [21] or principal directions [22]. However, they suffer from noisy, sparse, and non-uniform real-world point clouds with large-scale transformations. Recent advances include learning-based detectors such as USIP [23] that predicts repeatable keypoints by minimizing a probabilistic chamfer loss, and HRegNet [24] that further utilizes weighted farthest point sampling to select sparse keypoints from the predicted ones for hierarchical registration. 3DFeat-Net [25] extracts patch-wise descriptors with saliency scores for keypoint selection in a weakly supervised manner by minimizing a weighted feature alignment triplet loss. D3Feat [9] adopts a fully convolutional network to predict point-wise descriptors with hand-crafted saliency scores by minimizing a self-supervised detection loss.

3D Correspondence Learning. DCP [26] predicts soft correspondences from learned features and estimates the pose by a differential SVD layer. IDAM [27] designs iterative distance-aware similarity matrix convolution for iterative pairwise matching and pose estimation. Recently, coarse-to-fine correspondence learning has been regarded as a promising approach. The pioneering work CoFiNet [11] exploits a group of self-attention and cross-attention for coarse feature matching and the optimal transport for fine matching. GeoTransformer [12] proposes a geometric structure embedding for self-attention, and the local-to-global registration (LGR) for consistent pose estimation. RoITr [28] improves the coarse-to-fine framework with a rotation-invariant point cloud Transformer based on point pair features, while PEAL [29] and DiffusionPCR [30] use overlap priors and diffusion models for iterative feature matching, respectively. For outlier rejection, RANSAC [15] remains popular despite its inefficiency. DGR [31] predicts correspondence-wise confidence scores via a 6D convolutional network, while PointDSC [32] designs a consistency-guided non-local feature embedding to sample consistent correspondences for neural spectral matching and pose estimation.

3 Method

In this section, we present the proposed consistency-aware spot-guided Transformer (CAST) with a lightweight sparse-to-dense fine matching module for accurate and efficient point cloud registration.

3.1 Overview

Given two partially overlapped point clouds $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^3, i = 1, \cdots, M\}$ and $\mathbf{Y} = \{\mathbf{y}_j \in \mathbb{R}^3, j = 1, \cdots, N\}$, the point cloud registration problem can be formulated as solving the optimal rigid transformation between $\mathbf{X}, \mathbf{Y}$ by minimizing the weighted sum of point-to-point errors of a predicted correspondence set $\mathcal{C}$ with a confidence weight w_k for each correspondence $(\mathbf{x}_k, \mathbf{y}_k)$:

$$\min_{\mathbf{R},\mathbf{t}} \sum_{(\mathbf{x}_k,\mathbf{y}_k)\in\mathcal{C}} w_k \|\mathbf{R}\mathbf{x}_k + \mathbf{t} - \mathbf{y}_k\|_2^2, \tag{1}$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ are the rotation and the translation between $\mathbf{X}$ and $\mathbf{Y}$, respectively.

As depicted in Figure 1, CAST follows a coarse-to-fine feature matching and registration architecture, including a feature pyramid network, a consistency-aware spot-guided attention-based coarse matching module, and a sparse-to-dense fine matching module. We first utilize a KPConv-based fully convolutional network [33] to extract multi-scale features. We denote feature maps of the decoder with the size of $1/k$ as $\mathbf{F}^{1/k} = \{\mathbf{F}_X^{1/k}, \mathbf{F}_Y^{1/k}\}$, which correspond to nodes $\mathbf{X}^{1/k}$ and $\mathbf{Y}^{1/k}$ down-sampled from $\mathbf{X}, \mathbf{Y}$, respectively. For coarse matching, we first adopt an efficient linear cross-attention [34] module to enhance $\mathbf{F}^{1/4}$. Then both *semi-dense features* $\mathbf{F}^{1/4}$ and *coarse features* $\mathbf{F}^{1/8}$ are fed into a consistency-aware spot-guided attention-based coarse matching module to improve the feature distinctiveness. The similarity matrix $\mathbf{S} \in \mathbb{R}^{M'\times N'}$ between these enhanced semi-dense features $\hat{\mathbf{F}}_X \in \mathbb{R}^{M'\times D}, \hat{\mathbf{F}}_Y \in \mathbb{R}^{N'\times D}$ is computed based on inner product: $\mathbf{S} = \hat{\mathbf{F}}_X \hat{\mathbf{F}}_Y^{\mathsf{T}}$. Furthermore, we fed $\hat{\mathbf{F}}_X$ and $\hat{\mathbf{F}}_Y$ into a point-wise MLP to predict the overlap scores, which encode the likelihood of a node having a correspondence. We perform dual-softmax on $\mathbf{S}$ to obtain the final matching scores:

$$\mathbf{P}_{ij} = \hat{o}_i^X \hat{o}_j^Y \underset{k\in\{1,\cdots,M'\}}{\text{softmax}}(\mathbf{S}_{kj})_i \underset{k\in\{1,\cdots,N'\}}{\text{softmax}}(\mathbf{S}_{ik})_j, \tag{2}$$

where $\hat{o}_i^X$ and $\hat{o}_j^Y$ are predicted overlap scores of the *i*-th node of $\mathbf{X}^{1/4}$ and the *j*-th node of $\mathbf{Y}^{1/4}$, respectively. We use the mutual nearest neighbor scheme to select confident coarse correspondences.

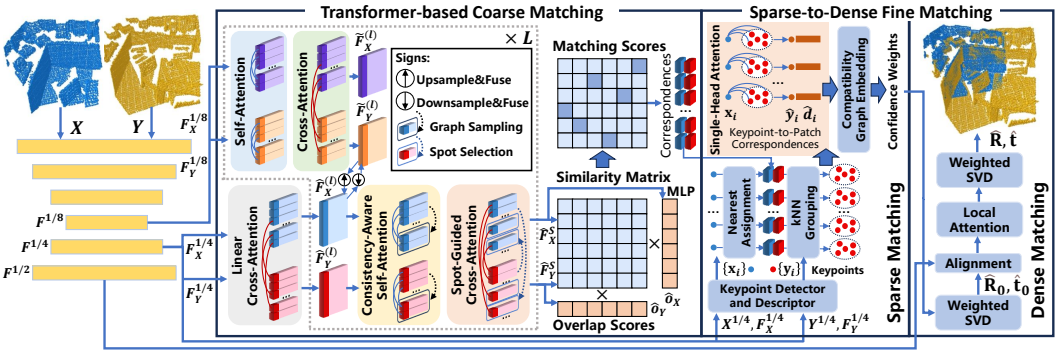

Figure 1: Overview of CAST. The feature pyramid network down-samples the point clouds and learns features in multiple resolutions. The coarse matching module extracts consistency-aware semi-dense correspondences via a group of alternate consistency-aware self-attention modules and spot-guided cross-attention modules with multi-scale feature fusion. Finally, the fine matching module predicts correspondences for both sparse keypoints and dense features and estimates the transformation.

For efficient fine matching, we extract a keypoint from the neighborhood of each semi-dense node in $\mathbf{X}^{1/4}$, and predict its virtual correspondence in $\mathbf{Y}^{1/4}$ based on the lightweight single-head attention. Then we utilize compatibility graph embedding to predict the confidence of these keypoint correspondences as weights in Eq. 1 for initial pose estimation. Finally, a lightweight local attention module for dense points $\mathbf{X}^{1/2}$ and $\mathbf{Y}^{1/2}$ predicts dense correspondences to refine the pose.

3.2 Consistency-Aware Spot-Guided Attention

To tackle the sparsity and looseness of coarse matching, we focus on feature aggregation among semi-dense features $\mathbf{F}^{1/4}$ leveraging both local and global geometric consistency. To be specific, the self-attention only attends to salient nodes sampled from a global compatibility graph, while the cross-attention only attends to nodes sampled based on local consistency, which are referred to as *consistency-aware self-attention* and *spot-guided cross-attention*, respectively.

Preliminaries. Transformers stacked by alternate self-attention and cross-attention have showcased advanced performance in coarse feature matching. When D-dimensional features $\mathbf{F}_A$ attends to $\mathbf{F}_B$, the output of vanilla attention is formulated as:

$$\hat{\mathbf{F}}_A = \text{softmax}\left(\frac{1}{\sqrt{D}}\mathbf{F}_A\mathbf{W}_Q(\mathbf{F}_B\mathbf{W}_K)^\mathsf{T}\right)\mathbf{F}_B\mathbf{W}_V, \tag{3}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are learnable linear transformations to generate queries, keys, and values. When $\mathbf{F}_A, \mathbf{F}_B$ related to coordinates $\mathbf{P}_A, \mathbf{P}_B$ are from the same point cloud, it becomes self-attention that requires positional encoding to embed spatial information. To encode the 3D relative positions, we equip the rotary positional embedding [35] $\tilde{\mathbf{R}}(\cdot)$ with learnable weights $\mathbf{b}_1, \cdots, \mathbf{b}_{D/2} \in \mathbb{R}^{1\times 3}$:

$$\tilde{\mathbf{R}}(\mathbf{p}) = \begin{bmatrix} \mathbf{R}(\mathbf{b}_1\mathbf{p}) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{R}(\mathbf{b}_{D/2}\mathbf{p}) \end{bmatrix}, \mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \forall \mathbf{p} \in \mathbb{R}^3. \tag{4}$$

When applying $\tilde{\mathbf{R}}(\cdot)$ to vanilla self-attention, the output is formulated as:

$$\hat{\mathbf{F}}_A = \text{softmax}\left(\frac{1}{\sqrt{D}}\mathbf{F}_A\mathbf{W}_Q\tilde{\mathbf{R}}(\mathbf{P}_A)(\mathbf{F}_B\mathbf{W}_K\tilde{\mathbf{R}}(\mathbf{P}_B))^\mathsf{T}\right)\mathbf{F}_B\mathbf{W}_V. \tag{5}$$

Architecture. As both spot-guided cross-attention and consistency-aware self-attention are sparse attention lacking of abundant global context, we propose to enhance the semi-dense features via multi-scale feature fusion with coarse features. Hence, the architecture of our coarse matching module is designed as a sequence of blocks for attention-based multi-scale feature aggregation. For each

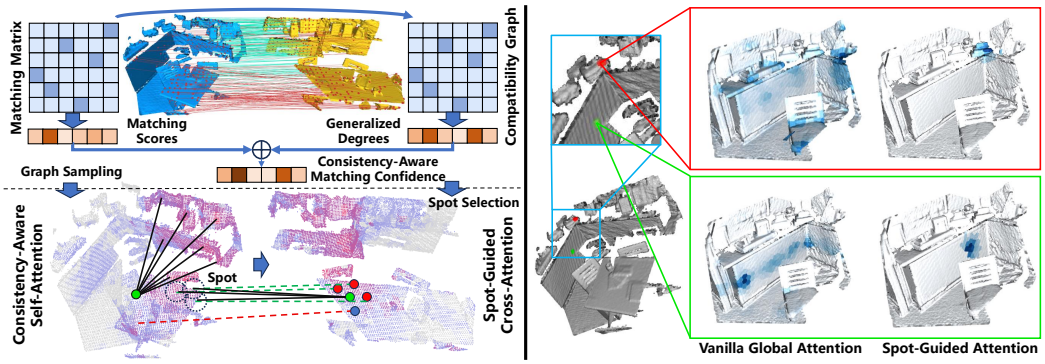

Figure 2: Illustration of consistency-aware self-attention and spot-guided cross-attention (Left), as well as visualization of the global cross-attention and spot-guided cross-attention (Right). For the left part, the green nodes are query nodes, while the red ones with correct correspondences (green dot lines) are reliable neighbors, and the blue one with a false correspondence (red dot line) is an unreliable neighbor. The self-attention (black lines) only attends to salient nodes while the cross-attention (black lines) only attends to spots (nodes within black circles).

block with both semi-dense features $\mathbf{F}^{1/4}$ and coarse features $\mathbf{F}^{1/8}$ as inputs, we first feed $\mathbf{F}^{1/8}$ into a self-attention module (Eq. 5) and a cross-attention module (Eq. 3). Then $\mathbf{F}^{1/4}$ and $\mathbf{F}^{1/8}$ are fused into each other based on nearest up-sampling and distance-based interpolated down-sampling [18]:

$$\begin{aligned}\hat{\mathbf{F}}^{1/4} &= \mathbf{F}^{1/4} + \text{MLP}(\text{Nearest Up-sampling}(\mathbf{F}^{1/8})),\\ \hat{\mathbf{F}}^{1/8} &= \mathbf{F}^{1/8} + \text{MLP}(\text{Interpolated Down-sampling}(\mathbf{F}^{1/4})).\end{aligned} \tag{6}$$

Finally, $\hat{\mathbf{F}}^{1/4}$ is fed into a consistency-aware self-attention module and a spot-guided cross-attention module at the end of each block. Before these sparse attention modules, we need to match the semi-dense features and evaluate the geometric consistency as a clue to select sparse yet instructive tokens. Given semi-dense features $\hat{\mathbf{F}}_X^{(l)}, \hat{\mathbf{F}}_Y^{(l)}$ in the l-th block, the matching score is formulated as:

$$\mathbf{P}_{ij}^{(l)} = \underset{k\in\{1,\cdots,M'\}}{\text{softmax}}(\mathbf{S}_{kj}^{(l)})_i \underset{k\in\{1,\cdots,N'\}}{\text{softmax}}(\mathbf{S}_{ik}^{(l)})_j,\ \mathbf{S}^{(l)} = \hat{\mathbf{F}}_X^{(l)}(\hat{\mathbf{F}}_Y^{(l)})^{\mathsf{T}}. \tag{7}$$

Then the correspondence of each node can be obtained as the node from another point cloud with the highest matching score, forming a correspondence set $\mathcal{C}^{(l)} = \{(\mathbf{x}_i^S, \mathbf{y}_i^S) : \mathbf{x}_i^S \in \mathbf{X}^{1/4}, \mathbf{y}_i^S \in \mathbf{Y}^{1/4}\}$. An insight about the consistency among correspondences is that the distance between two points is invariant after transformation. Hence, geometric compatibility is adopted as a simple yet effective measure of consistency [32], which is based on the length difference between pairwise line segments. Given a pre-defined threshold σ_c, the pair-wise geometric compatibility of $\mathcal{C}^{(l)}$ is formulated as:

$$\beta_{ij} = \left[1 - d_{ij}^2/\sigma_c^2\right]^+, d_{ij} = \left|\|\mathbf{x}_i^S - \mathbf{x}_j^S\|_2 - \|\mathbf{y}_i^S - \mathbf{y}_j^S\|_2\right|. \tag{8}$$

The compatibility matrix $\mathbf{B}_c = [\beta_{ij}]_{M'\times N'}$ is also considered as the adjacency matrix of a weighted undirected graph known as the compatibility graph, where each vertex is a pair of correspondence and the edge connectivity corresponds to the compatibility between two correspondences. Intuitively, we adopt the generalized degree of a pair of correspondence in the graph as a measure of global consistency, which quantifies the connectivity of a vertex as the sum of edge weights connected to it.

Consistency-Aware Self-Attention. Intuitively, the correspondences of less salient nodes can be effectively located based on the geometric relationships between them and the salient ones. Hence, compared with global self-attention that attends to all nodes, attending to only salient nodes is more efficient and effective to encode the geometric context for matching. We propose the consistency-aware self-attention that samples sparse salient nodes to be attended to based on both geometric consistency and feature similarity. Given the correspondence set $\mathcal{C}^{(l)}$ with a compatibility graph, we perform two-stage sampling by ranking the generalized degrees and matching scores, respectively. The first-stage graph sampling using generalized degrees can obtain sufficient consistent correspondences as proposals. The second-stage sampling based on matching scores can further obtain sparse salient nodes from these proposals. Finally, semi-dense features $\hat{\mathbf{F}}_X^{(l)}, \hat{\mathbf{F}}_Y^{(l)}$ only attend to features of salient nodes from the same point cloud for feature aggregation according to Eq. 5.

Spot-Guided Cross-Attention. As shown in Figure 2, global cross-attention tends to aggregate features from many irrelevant regions with similar patterns, leading to false correspondences. Inspired by local consistency that the correspondences of adjacent 3D points remain close to each other, we design the spot-guided cross-attention as depicted in Figure 2. For each node $\mathbf{x}_i^S$ such that $(\mathbf{x}_i^S, \mathbf{y}_i^S) \in \mathcal{C}^{(l)}$, we select a subset $\mathcal{N}_s(\mathbf{x}_i^S)$ from its neighborhood $\mathcal{N}(\mathbf{x}_i^S)$ as seeds, and construct a region of interest for it as $\mathcal{S}(\mathbf{x}_i^S) = \bigcup_{\mathbf{x}_k^S \in \mathcal{N}_s(\mathbf{x}_i^S)} \mathcal{N}(\mathbf{y}_k^S)$, namely its *spot*. $\mathcal{N}_s(\mathbf{x}_i^S)$ selects $\mathbf{x}_i^S$ and only its neighbors with reliable correspondences. We propose a consistency-aware matching confidence criterion to rank the neighbors, which is formulated as the product of the matching score and the normalized generalized degree in the compatibility graph. This criterion incorporates feature similarity and geometric consistency to properly measure the reliability of correspondences for seed selection. Finally, semi-dense features attend to their spots for feature aggregation according to Eq. 3. Under the guarantee of local consistency, the spots are likely to cover the true correspondences, providing guidance for feature aggregation without interfering with irrelevant areas.

3.3 Sparse-to-Dense Fine Matching

Given a coarse correspondence set $\hat{\mathcal{C}} = \{(\mathbf{x}_j^S, \mathbf{y}_j^S) : \mathbf{x}_j^S \in \mathbf{X}^{1/4}, \mathbf{y}_j^S \in \mathbf{Y}^{1/4}\}$ selected as mutual nearest neighbors from the final coarse matching scores (Eq. 2), we propose a lightweight sparse-to-dense fine matching module for hierarchical pose estimation without optimal transport, maintaining scalability and efficiency. For sparse matching, we first search *k*-nearest neighbors (kNN) of semi-dense nodes $\mathbf{X}^{1/4}$ among dense points $\mathbf{X}^{1/2}$ to group patches, then we use an attentive keypoint detector [36] to predict a repeatable keypoint with a descriptor from each patch. Each keypoint of point cloud $\mathbf{X}$ is assigned to its nearest node, and each node $\mathbf{y}_j^S \in \mathbf{Y}^{1/4}$ with a correspondence in $\hat{\mathcal{C}}$ groups a patch $\mathcal{P}(\mathbf{y}_j^S)$ of keypoints via kNN. Then, a keypoint $\mathbf{x}_i$ assigned to $\mathbf{x}_j^S$ will correspond to the patch $\mathcal{P}(\mathbf{y}_j^S)$, forming a pair of keypoint-to-patch correspondence. Finally, we utilize a shared single-head attention layer for each keypoint-to-patch correspondence to predict virtual correspondences for keypoints. Denote the descriptor of $\mathbf{x}_i$ as d_i^X, the virtual correspondence $\hat{\mathbf{y}}_i$ with feature $\hat{d}_i^Y$ is predicted from keypoints $\mathbf{y}_{i_1}, \cdots, \mathbf{y}_{i_k}$ with top-*k* descriptor similarity in $\mathcal{P}(\mathbf{y}_j^S)$ as:

$$\hat{\mathbf{y}}_i = \sum_{j=1}^{k} \text{softmax}(d_i^X \overline{\mathbf{W}}_Q (d_{i_j}^Y \overline{\mathbf{W}}_K)^\mathsf{T}) \mathbf{y}_{i_k}, \hat{d}_i^Y = \sum_{j=1}^{k} \text{softmax}(d_i^X \overline{\mathbf{W}}_Q (d_{i_j}^Y \overline{\mathbf{W}}_K)^\mathsf{T}) d_{i_k}^Y, \quad (9)$$

where $d_{i_1}^Y, \cdots, d_{i_k}^Y$ are descriptors of $\mathbf{y}_{i_1}, \cdots, \mathbf{y}_{i_k}$, and $\overline{\mathbf{W}}_Q$ and $\overline{\mathbf{W}}_K$ are learnable weights. Inspired by PointDSC [32], we construct a compatibility graph $\mathbf{B}$ (Eq. 8) of sparse keypoint correspondences $\{(\mathbf{x}_i, \hat{\mathbf{y}}_i)\}$ for spatial consistency filtering via compatibility graph embedding:

$$\mathbf{E}^{(l+1)} = \text{softmax}\left(\frac{1}{\sqrt{D_e}} \mathbf{E}^{(l)} \mathbf{W}_Q^{(l)} (\mathbf{E}^{(l)} \mathbf{W}_K^{(l)})^\mathsf{T} \odot \mathbf{B}\right) \mathbf{E}^{(l)} \mathbf{W}_V^{(l)}, \ \mathbf{E}_i^{(0)} = \text{MLP}([\mathbf{x}_i, d_i^X, \hat{\mathbf{y}}_i, \hat{d}_i^Y]), \quad (10)$$

where $\mathbf{E}^{(l)}$ is the correspondence-wise embedding of the l-th layer with learnable weights $\mathbf{W}_Q^{(l)}$, $\mathbf{W}_K^{(l)}$, $\mathbf{W}_V^{(l)}$. Finally, the embedding is fed into an MLP to classify if a correspondence is an inlier. The predicted inlier confidences serve as the weights of keypoint correspondences for pose estimation formulated as Eq. 1, which can be analytically solved by weighted Kabsch algorithm [37]. It is noteworthy that the above process is really lightweight and scalable to large-scale registration tasks.

After aligning two point clouds based on sparse matching, we propose to refine the transformation based on dense matching. We still utilize local attention (Eq. 9) to predict the correspondences of dense points $\mathbf{X}^{1/2}$ from its neighbors in $\mathbf{Y}^{1/2}$ within a radius R_d, and we simply set the confidence weight of a correspondence with a distance d as $w = [1 - d/R_d]^+$. By solving Eq. 1 again with both sparse and dense correspondences, we can achieve more accurate pose estimation efficiently.

3.4 Loss Functions

Our loss function needs to supervise four modules, *i.e.*, keypoint detection, coarse matching, keypoint matching, and dense registration. For keypoint detection, we utilize the probabilistic chamfer loss [23] $\mathcal{L}_p$ to minimize the distances between the closest keypoints from the source and target point clouds after alignment under the ground-truth transformation. Please refer to [23] for details.

Coarse Matching. Given the ground-truth coarse correspondence set $\mathcal{C}$ with an overlap ratio o_{ij} for each correspondence $(i, j) \in \mathcal{C}$, we propose a spot matching loss $\mathcal{L}_s$ and a coarse matching loss $\mathcal{L}_c$ formulated as weighted cross entropy losses to supervise the layer-wise coarse matching scores $\mathbf{P}^{(l)}$ $(l = 1, 2, \cdots, L)$ and the final coarse matching scores $\mathbf{P}$, respectively:

$$\mathcal{L}_s = -\frac{1}{L}\sum_{l=1}^{L}\frac{1}{\sum_{(i,j)\in\mathcal{C}} o_{ij}}\sum_{(i,j)\in\mathcal{C}} o_{ij}\log \mathbf{P}_{ij}^{(l)}, \tag{11}$$

$$\mathcal{L}_c = -\frac{1}{\sum_{(i,j)\in\mathcal{C}} o_{ij}}\sum_{(i,j)\in\mathcal{C}} o_{ij}\log \mathbf{P}_{ij} - \frac{1}{|\mathcal{N}_X|}\sum_{k\in\mathcal{N}_X}\log(1-\hat{o}_k^X) - \frac{1}{|\mathcal{N}_Y|}\sum_{k\in\mathcal{N}_Y}\log(1-\hat{o}_k^Y), \tag{12}$$

where $\mathcal{N}_X$ and $\mathcal{N}_Y$ are sets of semi-dense nodes in point clouds $\mathbf{X}$ and $\mathbf{Y}$ without correspondences, respectively. Two nodes are considered as a pair of coarse correspondence only when their ground-truth overlap ratio is greater than 0. Assuming that the patch centered at a point $p \in \mathbb{R}^3$ is a spherical neighborhood of radius r, the overlapping ratio o_{ij} of patches centered at $p_i \in \mathbf{X}^S$ and $q_j \in \mathbf{Y}^S$ with ground-truth translation $\mathbf{t} \in \mathbb{R}^3$ and rotation $\mathbf{R} \in SO(3)$ can be calculated by:

$$o_{ij} = \frac{2\pi\int_{d/2}^{r}(r^2-h^2)dh}{4\pi r^3/3} = 1 - \frac{3d}{4r} + \frac{d^3}{16r^3},\ d = \max\{\|\mathbf{R}p_i + \mathbf{t} - q_i\|, 2r\}. \tag{13}$$

Keypoint Matching. As our keypoint matching module follows a three-stage pipeline including similarity calculation, correspondence prediction, and consistency filtering, it is reasonable to supervise these stages with three losses, respectively. Only valid keypoint-to-patch correspondences are supervised during training, *i.e.*, the distance of the keypoint x and its closest point p_x in the patch C_x is less than a pre-defined threshold $R_p > 0$, and the points whose distances from x are greater than a pre-defined threshold $R_n \geq R_p$ form a non-empty set $N_x \subset C_x$. We formulate the keypoint matching loss $\mathcal{L}_f$ as an InfoNCE loss [38] with symmetric learnable weights W, which aims at maximizing the similarity between descriptors d_x and d_{p_x} of true correspondences (x, p_x) and minimizing the similarity between descriptors d_x and d_{n_x} of false correspondences $(x, n_x), n_x \in N_x$.

$$\mathcal{L}_f = -\mathbb{E}_{(x,p_x,N_x)}\left[\log\frac{\exp(d_x^{\mathsf{T}}W d_{p_x})}{\exp(d_x^{\mathsf{T}}W d_{p_x}) + \sum_{n_x\in N_x}\exp(d_x^{\mathsf{T}}W d_{n_x})}\right]. \tag{14}$$

For correspondence prediction, we adopt a L_2 loss $\mathcal{L}_k = \mathbb{E}_{(x,\hat{y})}\|\mathbf{R}x + \mathbf{t} - \hat{y}\|_2$ for the predicted correspondences $\hat{y}$ of keypoints x from all valid keypoint-to-patch correspondences (x, C_x). For consistency filtering, we simply utilize a binary entropy loss $\mathcal{L}_i$ to supervise the confidence scores of all keypoint correspondences. The binary ground-truth label of a keypoint correspondence $(x, \hat{y})$ is 1 if and only if it is an inlier, *i.e.*, $\|\mathbf{R}x + \mathbf{t} - \hat{y}\|_2$ is less than a threshold $R_f > 0$.

Dense Registration. Given the translation $\hat{\mathbf{t}}$ and rotation $\hat{\mathbf{R}}$ estimated by dense registration, we adopt a translation loss $\mathcal{L}_t = \|\hat{\mathbf{t}} - \mathbf{t}\|_2$ and a rotation loss $\mathcal{L}_r = \|\hat{\mathbf{R}}^{\mathsf{T}}\mathbf{R} - \mathbf{I}_{3\times3}\|_F$ for supervision.

Finally, we formulate our loss as $\mathcal{L} = \mathcal{L}_p + \lambda_s\mathcal{L}_s + \lambda_c\mathcal{L}_c + \lambda_f\mathcal{L}_f + \lambda_k\mathcal{L}_k + \lambda_i\mathcal{L}_i + \lambda_t\mathcal{L}_t + \lambda_r\mathcal{L}_r$, where $\lambda_c, \lambda_s, \lambda_f, \lambda_k, \lambda_i, \lambda_t, \lambda_r$ are balancing weights.

4 Experiments

In this section, we evaluate our method on both outdoor LiDAR point cloud datasets KITTI [39], nuScenes [40], and the indoor RGBD point cloud dataset 3DMatch [16]. Our network is trained using an AdamW [41] optimizer with a batch size of 1, an initial learning rate of 1e-4, and a weight decay of 1e-4. The step scheduler decrease the learning rate to 90% every five steps, with gradients clipped at a norm of 0.5 during back propagation. Despite the complexity of the loss function, only one stage is needed for training. Our model is trained on an NVIDIA RTX 3090 GPU with an Intel Xeon CPU @2.90GHZ for 5, 40, and 3 epochs on 3DMatch, KITTI, and nuScenes, respectively, and we set $\lambda_f = \lambda_i = 1, \lambda_r = 20, \lambda_t = 5, \lambda_s = 0.1$, and $\lambda_c = 0.2, \lambda_k = 1$ for KITTI and nuScenes, $\lambda_c = 1, \lambda_k = 10$ for 3DMatch.

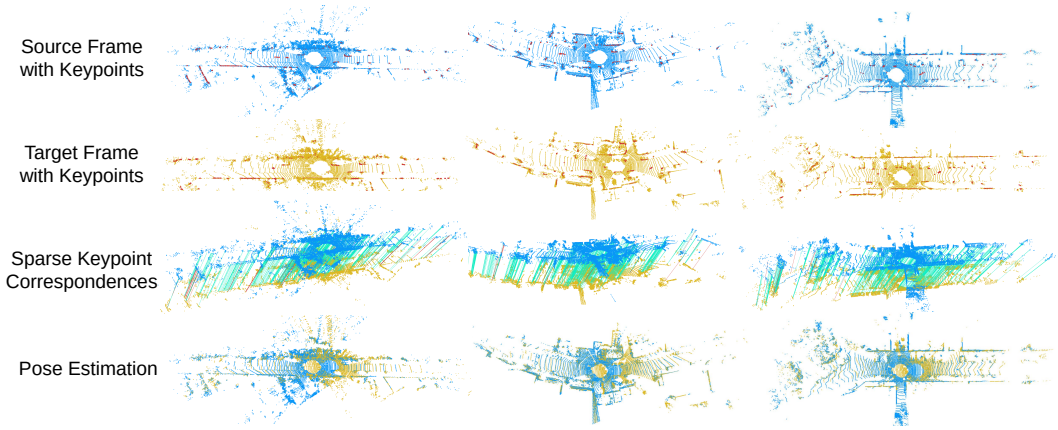

Figure 3: Qualitative registration results on KITTI dataset. We show three examples in three columns. The first two rows present the raw point clouds and highlight the 3D keypoints with low uncertainty in red. Our keypoints are typically located in sharp corners and edges of buildings, pillars, and vehicles. The third row shows the predicted sparse keypoint correspondences with high scores, while the last row presents the aligned point clouds after pose estimation. Although a few outliers colored in red have not been filtered out, their distances are acceptable for accurate registration.

4.1 Outdoor Scenarios: KITTI and NuScenes

KITTI [39] is a popular benchmark for autonomous driving. Following [9], we use sequences 0 to 5 for training, 6 to 7 for validation, and 8 to 10 for testing, and select only point cloud pairs at least 10m away from each other with ICP-refined [42] GPS localization results as ground truth. NuScenes [40] is another large-scale outdoor autonomous driving benchmark including 850 scenes for training and validation and 150 for testing. Following [24], we select each LiDAR keyframe with the second keyframe after it as a pair of point clouds. We use three metrics for evaluation [20]: *relative translation error* (RTE), *relative rotation error* (RRE), and *registration recall* (RR).

Table 1: Registration performance on KITTI odometry dataset.

Model	Publication	RTE (cm)	RRE (°)	RR (%)
3DFeat-Net	ECCV 2018 [25]	25.9	0.25	96.0
FCGF	ICCV 2019 [8]	9.5	0.30	96.6
D3Feat	CVPR 2020 [9]	7.2	0.30	99.8
SpinNet	CVPR 2021 [19]	9.9	0.47	99.1
Predator	CVPR 2021 [20]	6.8	0.27	99.8
CoFiNet	NeurIPS 2021 [11]	8.2	0.41	99.8
GeoTransformer	CVPR 2022 [12]	6.8	0.24	99.8
OIF-Net	NeurIPS 2022 [13]	6.5	**0.23**	99.8
PEAL	CVPR 2023 [29]	6.8	**0.23**	99.8
DiffusionPCR	CVPR 2024 [30]	6.3	**0.23**	99.8
MAC	CVPR 2023 [43]	8.5	0.40	99.5
RegFormer	ICCV 2023 [44]	8.4	0.24	99.8
CAST		**2.5**	0.27	**100.0**

Our results on KITTI are detailed quantitatively in Table 1 and qualitatively in Figure 3. Table 1 shows that the proposed CAST outperforms various learning-based methods, including descriptor-based [25, 8, 9, 19, 20], coarse-to-fine correspondence-based [11, 12, 13] including the latest ones with iterative matching [29, 30], a recent graph-based [43] and an end-to-end [44] baselines. Specifically, CAST achieves a RR of 100.0% and the lowest RTE of 2.5cm, which is 60.3% improvement over the *state-of-the-art* DiffusionPCR [30], highlighting its superior robustness and accuracy.

As for RRE, CAST slightly underperforms some coarse-to-fine methods [12, 13, 29, 30], primarily due to numerical errors in SVD-based pose estimation that usually produces non-orthonormal rotation matrices. RRE is set as 0 when $\text{trace}(\hat{\mathbf{R}}^\top \mathbf{R}) > 3$, a condition met frequently across all methods. Consequently, geodesic distance-based RRE may not accurately reflect the actual performance.

Table 3: Evaluation results on indoor RGBD point cloud datasets.

	Dataset	3DMatch					3DLoMatch					Average
		Registration Recall (%)					Registration Recall (%)					Time (s)
	Samples	5000	2500	1000	500	250	5000	2500	1000	500	250	All
descriptor-based	PerfectMatch [7]	78.4	76.2	71.4	67.6	50.8	33.0	29.0	23.3	17.0	11.0	-
	FCGF [8]	85.1	84.7	83.3	81.6	71.4	40.1	41.7	38.2	35.4	26.8	0.271
	D3Feat [9]	81.6	84.5	83.4	82.4	77.9	37.2	42.7	46.9	43.8	39.1	0.289
	SpinNet [19]	88.6	86.6	85.5	83.5	70.2	59.8	54.9	48.3	39.8	26.8	90.804
	YOHO [50]	90.8	90.3	89.1	88.6	84.5	65.2	65.5	63.2	56.5	48.0	13.529
	Predator [20]	89.0	89.9	90.6	88.5	86.6	59.8	61.2	62.4	60.8	58.1	0.759
correspondence-based	REGTR [47]			92.0					64.8			0.382
	CoFiNet [11]	89.3	88.9	88.4	87.4	87.0	67.5	66.2	64.2	63.1	61.0	0.306
	GeoTransformer [12]	92.0	91.8	91.8	91.4	91.2	75.0	74.8	74.2	74.1	73.5	0.192
	OIF-Net [13]	92.4	91.9	91.8	92.1	91.2	76.1	75.4	75.1	74.4	73.6	0.555
	RoITr [28]	91.9	91.7	91.8	91.4	91.0	74.7	74.8	74.8	74.2	73.6	0.457
	PEAL [29]	94.4	94.1	94.1	93.9	93.4	79.2	79.0	78.8	78.5	77.9	2.074
	BUFFER [48]			92.9					71.8			0.290
	SIRA-PCR [49]	93.6	93.9	93.9	92.7	92.4	73.5	73.9	73.0	73.4	71.1	0.291
	DiffusionPCR [30]	94.4	94.3	94.5	94.0	93.9	**80.0**	**80.4**	**79.2**	**78.8**	**78.8**	1.964
	CAST			**95.2**					75.1			**0.182**

For a more challenging LiDAR benchmark nuScenes, we compare CAST with both traditional [42, 45, 15] and learning-based algorithms [26, 27, 46, 31, 24] in Table 2. We do not include the coarse-to-fine methods since none have been trained or tested on nuScenes. Most of the results are borrowed from [24] while HRegNet [24] is re-evaluated with their open source codes. Our method achieves the lowest translation error of 0.12m and the lowest rotation error of 0.20° while maintaining the best RR of 99.9%, showcasing *state-of-the-art* robustness and accuracy.

Table 2: Registration performance on nuScenes.

Method	RTE (m)	RRE (°)	RR (%)
Point-to-Point ICP [42]	0.25	0.25	18.8
Point-to-Plane ICP [42]	0.15	0.21	36.8
FGR [45]	0.71	1.01	32.2
RANSAC [15]	0.21	0.74	60.9
DCP [26]	1.09	2.07	56.8
IDAM [27]	0.47	0.79	88.0
FMR [46]	0.60	1.61	92.1
DGR [31]	0.21	0.48	98.4
HRegNet [24]	0.18	0.45	**99.9**
CAST	**0.12**	**0.20**	**99.9**

4.2 Indoor Scenarios: 3DMatch and 3DLoMatch

Our approach is also evaluated on indoor benchmarks 3DMatch [16] and 3DLoMatch [20], which consist of point cloud pairs with overlaps >30% and $10\% \sim 30\%$, respectively. In Table 3, we use registration recall [16] as our evaluation metric, and test the runtime of all methods in Pytorch implementation with 5000 points on our device with an Intel CPU i7-12800HX@2.30GHZ and an NVIDIA RTX 3080Ti GPU for fairness, except [7] in Tensorflow implementation and [13, 29, 30] using the results reported in their papers [13, 30] due to the absence of source codes. Our method along with other sparse matching baselines [47, 48] directly uses all points for evaluation. To enhance the robustness in low overlapping cases, our method is combined with RANSAC estimating an initial pose from only 250 coarse correspondences to reject the outliers during fine matching.

On the 3DMatch benchmark, our method achieves *state-of-the-art* RR of 95.2%. On the more challenging 3DLoMatch, CAST achieves a high RR of 75.1%, outperforming all descriptors and non-iterative correspondence-based methods [47, 11, 12, 28, 48, 49] except OIF-Net [13] using more than 1000 sampled points. As our method typically detects about 1000 sparse keypoints and establishes less than 250 keypoint correspondences on 3DLoMatch, it is fair to compare CAST with other methods using only 250 sample points. However, CAST outperforms the *state-of-the-art* non-iterative correspondence-based methods OIF-Net [13] using less than 1000 points. Notably, our method achieves such superior performance only with the lowest runtime, while RANSAC remains efficient due to our high inlier ratio. Although PEAL [29] and DiffusionPCR [30] show higher RR on 3DLoMatch, their iterative feature matching with overlap priors is extremely time-consuming (10 times of ours), while PEAL even requires extra information from 2D images.

Table 4: Ablation studies of coarse matching modules on indoor datasets.

	MS	SG	CA	OV	3DMatch PIR (%)	3DMatch PMR (%)	3DMatch RR (%)	3DLoMatch PIR (%)	3DLoMatch PMR (%)	3DLoMatch RR (%)
1	✓			✓	77.56	95.87	94.45	40.82	70.58	72.07
2	✓	✓		✓	77.95	96.61	94.92	42.55	72.77	74.57
3		✓	✓	✓	69.58	96.67	94.14	32.59	65.02	73.00
4	✓	✓	✓		73.56	**97.17**	95.07	35.25	68.33	74.91
5	✓	✓	✓	✓	**79.79**	**97.17**	**96.01**	**44.41**	**75.24**	**76.59**

4.3 Ablation Studies

We select indoor datasets for ablation studies of coarse matching as they are more challenging. Here we evaluate the RR over the whole dataset rather than the average RR of eight sequences reported in Table 3, which is more reasonable for a dataset with significant variances of sequence lengths. Besides, we assess two extra metrics to directly measure the performance of coarse matching: *patch inlier ratio* (PIR), the fraction of patch matches with actual overlap; and *patch matching recall* (PMR), the fraction of point cloud pairs with PIR above 20%. Results from five experiments in Table 4 demonstrate the effects of the proposed multi-scale feature fusion (MS), spot-guided cross-attention (SG), consistency-aware self-attention (CA), and the overlap head for overlap score prediction (OV). The first experiment ablating CA and replacing SG with linear cross-attention, suffers performance degradation in all metrics due to inconsistency. The second experiment improves all metrics based on SG, while the last one achieves the best performance via CA, showcasing their effectiveness. Figure 2 visualizes the vanilla global cross-attention and our spot-guided cross-attention. Instead of interacting with many similar yet irrelevant regions for misleading feature aggregation, SG can effectively select instructive areas to attend to according to local consistency. Compared to the last experiment, the third one verifies the effectiveness of multi-scale feature fusion, while the fourth one demonstrates the necessity of overlap prediction.

Additionally, we conducted five ablation studies on KITTI for a better understanding of our fine matching, since pose errors are better metrics to reflect accuracy. The second experiment using only sparse keypoints for registration highlights the effectiveness of dense registration, while the third one shows the effect of learnable dense correspondences compared to nearest neighbors. The last three experiments report the performance of sparse registration by ablating the keypoint detector, the learnable sparse correspondences, and the compatibility graph embedding, each demonstrating their necessity for accuracy. Despite these variations, all studies maintain a 100% RR, showing the robustness of coarse matching.

Table 5: Ablation studies of fine matching on KITTI.

	RTE (cm)	RRE (°)	RR (%)
ours	**2.51**	**0.27**	**100.00**
ours w/o dense registration	3.13	0.30	**100.00**
ours w/o virtual dense corr.	2.85	0.28	**100.00**
ours w/o keypoint detection	3.58	0.30	**100.00**
ours w/o virtual sparse corr.	3.25	0.30	**100.00**
ours w/o graph embedding	5.01	0.30	**100.00**

5 Conclusion

In this paper, we present a novel consistency-aware spot-guided Transformer to achieve compact and consistent coarse matching for point cloud registration. At the coarse matching stage, our consistency-aware self-attention enhances the feature representations with sparse sampling from the geometric compatibility graph. Additionally, our spot-guided cross-attention leverages local consistency to guide the cross-attention to confident spots without interfering with irrelevant areas. Based on these semi-dense and consistent coarse correspondences, a lightweight and scalable sparse-to-dense fine matching module empowered by local attention can achieve accurate pose estimation without optimal transport or hypothesis-and-selection pipelines. Our method has showcased *state-of-the-art* accuracy, robustness, and efficiency for point cloud registration across different 3D sensors and scenarios, which paves the way for large-scale real-time applications such as SLAM.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62203383/62088101).

References

[1] Renlang Huang, Minglei Zhao, Jiming Chen, and Liang Li. Kdd-loam: Jointly learned keypoint detector and descriptors assisted lidar odometry and mapping. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8559–8565, 2024.

[2] Weitong Hua, Zhongxiang Zhou, Jun Wu, Huang Huang, Yue Wang, and Rong Xiong. Rede: End-to-end object 6d pose robust estimation using differentiable outliers elimination. *IEEE Robotics and Automation Letters*, 6(2):2886–2893, 2021.

[3] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8953–8962, 2021.

[4] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3384–3391. IEEE, 2008.

[5] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217. IEEE, 2009.

[6] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.

[7] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5545–5554, 2019.

[8] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8958–8966, 2019.

[9] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6359–6367, 2020.

[10] Jiahuan Yu, Jiahao Chang, Jianfeng He, Tianzhu Zhang, Jiyang Yu, and Feng Wu. Adaptive spot-guided transformer for consistent local feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21898–21908, 2023.

[11] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust point cloud registration. *Advances in Neural Information Processing Systems*, 34:23872–23884, 2021.

[12] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, Slobodan Ilic, Dewen Hu, and Kai Xu. Geotransformer: Fast and robust point cloud registration with geometric transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[13] Fan Yang, Lin Guo, Zhi Chen, and Wenbing Tao. One-inlier is first: Towards efficient position encoding for point cloud registration. *Advances in Neural Information Processing Systems*, 35:6982–6995, 2022.

[14] Minhao Li, Zheng Qin, Zhirui Gao, Renjiao Yi, Chenyang Zhu, Yulan Guo, and Kai Xu. 2d3d-matr: 2d-3d matching transformer for detection-free registration between images and point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14128–14138, 2023.

[15] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[16] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1802–1811, 2017.

[17] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–205, 2018.

[18] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[19] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11753–11762, 2021.

[20] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4267–4276, 2021.

[21] Hui Chen and Bir Bhanu. 3d free-form object recognition in range images using local surface patches. *Pattern Recognition Letters*, 28(10):1252–1262, 2007.

[22] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *IEEE International Conference on Computer Vision Workshops, ICCV workshops*, pages 689–696. IEEE, 2009.

[23] Jiaxin Li and Gim Hee Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 361–370, 2019.

[24] Fan Lu, Guang Chen, Yinlong Liu, Lijun Zhang, Sanqing Qu, Shu Liu, Rongqi Gu, and Changjun Jiang. Hregnet: A hierarchical network for efficient and accurate outdoor lidar point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[25] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 607–623, 2018.

[26] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2019.

[27] Jiahao Li, Changhao Zhang, Ziyao Xu, Hangning Zhou, and Chi Zhang. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 378–394. Springer, 2020.

[28] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5384–5393, 2023.

[29] Junle Yu, Luwei Ren, Yu Zhang, Wenhui Zhou, Lili Lin, and Guojun Dai. Peal: Prior-embedded explicit attention learning for low-overlap point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17702–17711, 2023.

[30] Zhi Chen, Yufan Ren, Tong Zhang, Zheng Dang, Wenbing Tao, Sabine Süsstrunk, and Mathieu Salzmann. Diffusionpcr: Diffusion models for robust multi-step point cloud registration. *arXiv preprint arXiv:2312.03053*, 2023.

[31] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2514–2523, 2020.

[32] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. Pointdsc: Robust point cloud registration using deep spatial consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15859–15869, 2021.

[33] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019.

[34] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020.

[35] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[36] Fan Lu, Guang Chen, Yinlong Liu, Zhongnan Qu, and Alois Knoll. Rskdd-net: Random sample-based keypoint detector and descriptor. *Advances in Neural Information Processing Systems*, 33:21297–21308, 2020.

[37] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[38] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[39] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[40] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[42] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[43] Xiyu Zhang, Jiaqi Yang, Shikun Zhang, and Yanning Zhang. 3d registration with maximal cliques. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17745–17754, 2023.

[44] Jiuming Liu, Guangming Wang, Zhe Liu, Chaokang Jiang, Marc Pollefeys, and Hesheng Wang. Regformer: An efficient projection-aware transformer network for large-scale point cloud registration. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8417–8426, 2023.

[45] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 766–782. Springer, 2016.

[46] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11366–11374, 2020.

[47] Zi Jian Yew and Gim Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6677–6686, 2022.

[48] Sheng Ao, Qingyong Hu, Hanyun Wang, Kai Xu, and Yulan Guo. Buffer: Balancing accuracy, efficiency, and generalizability in point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1255–1264, 2023.

[49] Suyi Chen, Hao Xu, Ru Li, Guanghui Liu, Chi-Wing Fu, and Shuaicheng Liu. Sira-pcr: Sim-to-real adaptation for 3d point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14394–14405, 2023.

[50] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1630–1641, 2022.

[51] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

[52] Kevin Lai, Liefeng Bo, and Dieter Fox. Unsupervised feature learning for 3d scene labeling. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3050–3057. IEEE, 2014.

[53] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 323–332. IEEE, 2016.

[54] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017.

[55] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1625–1632, 2013.

[56] Maciej Halber and Thomas Funkhouser. Fine-to-coarse global registration of rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1755–1764, 2017.

[57] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 303–312, 1996.

[58] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2021.

[59] Zhi Chen, Kun Sun, Fan Yang, and Wenbing Tao. Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13221–13231, 2022.

[60] Yifei Zhang, Hao Zhao, Hongyang Li, and Siheng Chen. Fastmac: Stochastic spectral sampling of correspondence graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17857–17867, 2024.

[61] François Pomerleau, M. Liu, Francis Colas, and Roland Siegwart. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31(14):1705–1711, December 2012.

[62] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 1482–1489, 2005.

A Appendix

In this appendix, we will first detail our neural network architecture with hyper-parameters in Sec. A.1. Then we introduce the related datasets in Sec. A.2 and provide more quantitative experimental results of our method with detailed explanation of evaluation metrics in Sec. A.3. Additionally, more qualitative results are illustrated in Sec. A.4. Finally, we will discuss the limitations and broader impacts in Sec. A.5 and Sec. A.6, respectively.

A.1 Neural Network Architectures and Hyper-parameters

Feature Pyramid Network. CAST utilizes a fully convolutional feature pyramid network as the backbone, which follows an encoder-decoder architecture based on KPConv [33] operations. Details of our network architecture are illustrated in Figure 4, which remain the same as [12] including five encoder layers and three decoder layers. Note that the backbone for indoor datasets 3DMatch [16] is slightly different, which only comprises four encoder layers and two decoder layers.

Coarse Matching Module. Figure 5 illustrates the details of our consistency-aware spot-guided attention blocks for coarse matching. Both coarse features and semi-dense features extracted from the backbone are first projected to 128 dimensions and then pass through three consistency-aware spot-guided attention blocks (Figure 1). Each attention module uses 4 heads with ReLU activation. Compared to vanilla attention with a quadratic increase in the size of the attention matrix with respect to the input length, the linear attention [34] is much more efficient in global context aggregation by replacing the softmax operator with the product of two kernel functions:

$$\text{Linear attention}(Q, K, V) = \phi(Q)(\phi(K)^{\mathsf{T}}V), \tag{15}$$

where $\phi(\cdot) = \text{elu}(\cdot) + 1$. For spot-guided cross-attention, each node $\mathbf{x}_i^S$ selects 4 seeds as $\mathcal{N}_s(\mathbf{x}_i^S)$ from the neighborhood $\mathcal{N}(\mathbf{x}_i^S)$ with 12 nodes based on consistency-aware matching confidence. For consistency-aware self-attention, we first scale the generalized degrees to $[0, 1]$, and sample 48 nodes with highest matching scores from nodes with scaled degrees greater than 0.3 as keys to be attended.

Fine Matching Modules. For keypoint detection and description, we utilize an attentive keypoint detector [36] to extract a keypoint with a descriptor from each local patch, which contains $k = 32$ nearest neighbors among dense points $\mathbf{x}_{i_1}, \cdots, \mathbf{x}_{i_k} \in \mathbf{X}^{1/2}$ with features $f_{i_1}, \cdots, f_{i_k}$ of a patch

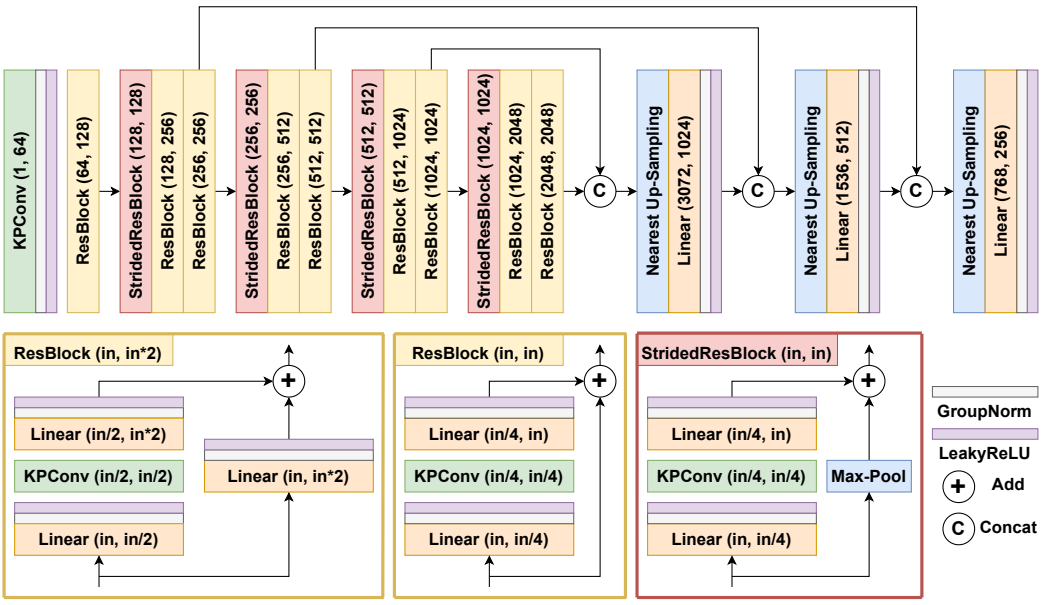

Figure 4: The detailed architecture of the KPConv-based feature pyramid network.

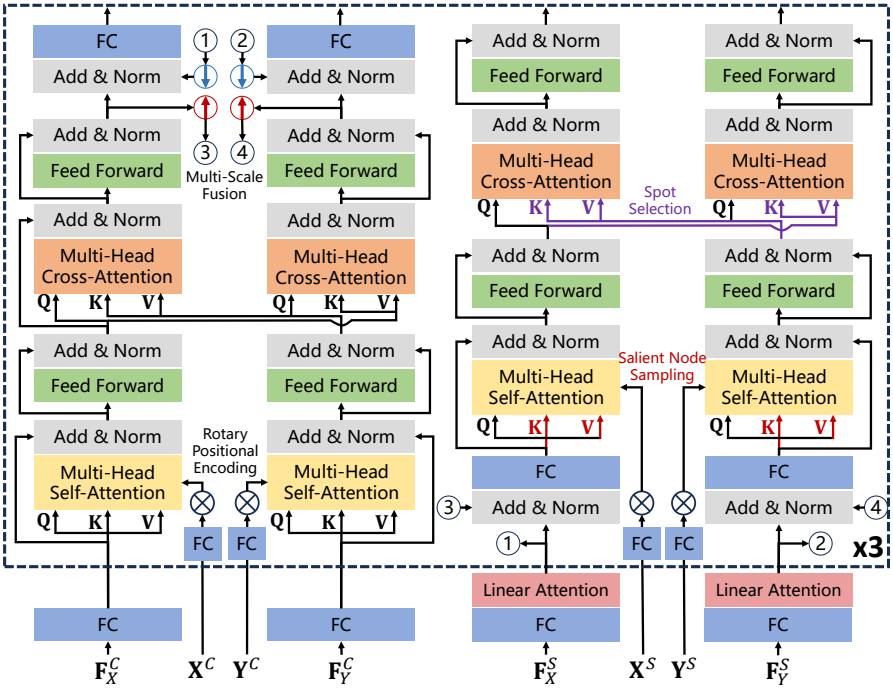

Figure 5: The detailed architecture of the consistency-aware spot-guided Transformer with multi-scale feature fusion for coarse feature matching.

node $\mathbf{x}_i^S \in \mathbf{X}^S$. The details of the keypoint detector and descriptor are illustrated in Figure 6, which are based on only shared MLPs with $\left[\mathbf{x}_{i_j}, \|\mathbf{x}_{i_j} - \mathbf{x}_i^S\|_2, f_{i_j}\right], j = 1, \cdots, k$ as inputs. For keypoints $\mathbf{x}_1^K, \cdots, \mathbf{x}_{M'}^K$ with uncertainties $\sigma_1, \cdots, \sigma_{M'}$ and $\mathbf{y}_1^K, \cdots, \mathbf{y}_{N'}^K$ with uncertainties $\sigma_1, \cdots, \sigma_{N'}$ from point cloud $\mathbf{X}$ and $\mathbf{Y}$, respectively, we adopt the probabilistic chamfer loss in [23] for training, please refer to their paper for details.

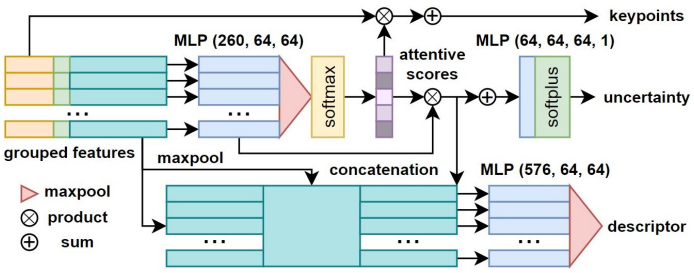

Figure 6: Attentive keypoint detector and descriptor.

To establish keypoint-to-patch correspondences based on coarse correspondences, each keypoint is assigned to its nearest node when their distance is below a threshold R_k. Then the keypoint corresponds to the corresponding patch of this node containing k_p keypoints. As mentioned in Sec. 3, we leverage a single-head attention layer to predict the virtual correspondence of each keypoint based on only k_s keypoints in its corresponding patch with the highest similarity (Eq. 9). Finally, spatial consistency filtering is performed via three graph compatibility embedding layers with embedding dimension $D_e = 64$ in Eq. 10. After sparse matching and registration, we further refine the transformation based on dense matching, which searches k_d nearest neighbors within a distance threshold R_d for each dense point to predict its virtual dense correspondence. Specifically, we set $k_s = 4, k_d = 6$ in our implementation, while other related hyper-parameters are listed in Table 6.

Weighted Kabsch Algorithm. The weighted Kabsch algorithm [37], also known as the Proscrutes algorithm, provides the closed-form solution for the point cloud registration problem (Eq. 1). Given a predicted correspondence set $\hat{\mathcal{C}} = \{(\mathbf{x}_k, \mathbf{y}_k) : k = 1, \cdots, n\}$, the optimal rigid transformation $\hat{\mathbf{R}} \in SO(3), \hat{\mathbf{t}} \in \mathbb{R}^3$ can be estimated via two steps:

Table 6: Hyper-parameter setting for three datasets in our implementation.

Hyper-parameters	Explanation	3DMatch	KITTI	nuScenes
r	coarse overlap radius (Eq. 13)	0.075m	1.2m	1.2m
σ_c	coarse compatibility threshold (Eq. 8)	0.15m	1.8m	1.8m
R_k	keypoint-to-node distance threshold	0.1m	1.8m	1.8m
σ_d	fine compatibility threshold	0.1m	1.0m	1.0m
R_d	dense matching radius	0.15m	0.75m	1.0m
R_p	positive matching threshold (Eq. 14)	0.05m	0.45m	0.45m
R_n	negative matching threshold (Eq. 14)	0.06m	0.6m	0.6m
k_p	number of keypoints in a patch	16	24	24

Step 1. Centralize the point clouds by subtracting away their weighted centroids:

$$\tilde{\mathbf{x}}_k = \mathbf{x}_k - \overline{\mathbf{x}},\ \tilde{\mathbf{y}}_k = \mathbf{y}_k - \overline{\mathbf{y}},\ \overline{\mathbf{x}} = \frac{\sum_{k=1}^{n} w_k \mathbf{x}_k}{\sum_{k=1}^{n} w_k},\ \overline{\mathbf{y}} = \frac{\sum_{k=1}^{n} w_k \mathbf{y}_k}{\sum_{k=1}^{n} w_k}. \tag{16}$$

Step 2. Estimate the transformation. A 3×3 weighted covariance matrix can be computed as

$$\mathbf{H} = \sum_{k=1}^{n} w_k \tilde{\mathbf{x}}_k \tilde{\mathbf{y}}_k^{\mathsf{T}}. \tag{17}$$

With its singular value decomposition $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{T}}$, the optimal estimate of pose is given by:

$$\hat{\mathbf{R}} = \mathbf{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V}\mathbf{U}^{\mathsf{T}}) \end{bmatrix} \mathbf{U}^{\mathsf{T}},\ \hat{\mathbf{t}} = \overline{\mathbf{y}} - \hat{\mathbf{R}}\overline{\mathbf{x}}. \tag{18}$$

A.2 Data and Benchmarks

3DMatch. 3DMatch [16] is a widely used RGBD point cloud dataset for 3D reconstruction, including 62 scenes from 7-Scenes [51], RGB-D Scenes V2 [52], Analysis-by-Synthesis [53], BundleFusion [54], SUN3D [55] and Halber *et al.* [56] with their licenses in Table 7, where 46 scenes are used for training, 8 scenes for validation and 8 scenes for testing. Input point clouds are generated by fusing 50 consecutive depth frames collected by RGBD cameras using TSDF volumetric fusion [57]. Different from the original 3DMatch [16] that only consists of point cloud pairs with >30% overlaps, [20] also includes point cloud pairs with overlaps between 10% and 30% for training, and it sets two benchmarks for performance evaluation, *i.e.*, 3DMatch consisting of point cloud pairs with >30% overlaps, and 3DLoMatch consisting of point cloud pairs with low overlap ratios between 10% and 30%. Following [12], we utilize the voxel grid down-sampling of 2.5cm voxel size for data preprocessing, which picks the centroid as a down-sampled point when multiple points fall into a common voxel grid. We leverage the data and evaluation protocols in [20] for training and testing.

Table 7: Raw data in the 3DMatch [16] dataset and their licenses.

Datasets	License
7-Scenes [51]	Non-commercial use only
Analysis-by-Synthesis [53]	CC BY-NC-SA 4.0
BundleFusion [54]	CC BY-NC-SA 4.0
RGB-D Scenes v2 [52]	(License not stated)
SUN3D [55]	CC BY-NC-SA 4.0
Halber *et al.* [56]	CC BY-NC-SA 4.0

KITTI. KITTI [39] is a classic benchmark under the NonCommercial-ShareAlike 3.0 License for a variety of computer vision tasks of autonomous driving, ranging from LiDAR-based or vision-based or multi-sensors based odometry, object detection and tracking, optical flow estimation, point cloud registration, *etc*. KITTI comprises of 11 sequences scanned by a Velodyne HDL-64 3D laser scanner in driving scenarios. Following the data splitting method in [8], we use sequences 0 to 5 for training, 6 to 7 for validation, and 8 to 10 for testing. Besides, we directly leverage the source code of [9] to

select point cloud pairs which are at least 10m away from each other, which leads to 1,358 training pairs, 180 validation pairs, and 555 testing pairs. Moreover, as the ground truth transformations provided by GPS are less accurate, we follow [9] to refine them via standard ICP [42] in 500 iterations. Following [9], we utilize the voxel down-sampling of 0.3m voxel size for data preprocessing.

NuScenes. NuScenes [40] is an outdoor autonomous driving datasets under CC BY-NC-SA 4.0 license. It is the first large-scale dataset to provide data from the entire sensor suite of an autonomous vehicle, consisting of 850 scenes for training and validation and 150 scenes for testing. Following [24], we select the first 700 scenes from the 850 scenes for training and the others for validation. The information about the point cloud pairs with ground-truth transformations is downloaded from the source codes of [24] which selects each LiDAR keyframe with the second keyframe after it as a pair. For data preprocessing, we apply 0.3m voxel grid down-sampling.

A.3 Evaluation Metrics with Extra Quantitative Results

The outdoor datasets KITTI [39] and nuScenes [40] commonly use three metrics for evaluation [20]: (1) *relative translation error* (RTE), the Euclidean distance between the estimated and ground-truth translation vectors; (2) *relative rotation error* (RRE), the geodesic distance between the estimated and ground-truth rotation matrices on $SO(3)$; and (3) *registration recall* (RR), the percentage of point cloud pairs with RTE<2m and RRE<5°. Given the ground-truth rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$, as well as the estimated rotation $\hat{\mathbf{R}}$ and translation $\hat{\mathbf{t}}$, the RRE and RTE are formulated as

$$\text{RRE} = \arccos\left(\frac{\text{trace}(\hat{\mathbf{R}}^\top \mathbf{R}) - 1}{2}\right), \ \text{RTE} = \|\hat{\mathbf{t}} - \mathbf{t}\|_2. \tag{19}$$

The indoor benchmarks 3DMatch [16] and 3DLoMatch [20] commonly leverage three metrics for evaluation: *registration recall* (RR), *inlier ratio* (IR), and *feature matching recall* (FMR). To keep with existing methods [20], we exclude the consecutive point clouds when evaluating the RR.

Registration Recall refers to the percentage of point cloud pairs whose root mean square error (RMSE) is less than a pre-defined threshold $\tau = 0.2$m. Given point clouds $\mathbf{X}, \mathbf{Y}$ with a ground-truth correspondence set $\mathbf{C} = \{(\mathbf{x}_i, \mathbf{y}_j) : \mathbf{x}_i \in \mathbf{X}, \mathbf{y}_j \in \mathbf{Y}\}$, estimated rotation $\hat{\mathbf{R}} \in SO(3)$, and estimated translation $\hat{\mathbf{t}} \in \mathbb{R}^3$, the RMSE of $\mathbf{X}, \mathbf{Y}$ is formulated as

$$\text{RMSE}(\mathbf{X}, \mathbf{Y}) = \sqrt{\frac{1}{|\mathbf{C}|} \sum_{(\mathbf{x}_i, \mathbf{y}_j) \in \mathbf{C}} \|\hat{\mathbf{R}}\mathbf{x}_i + \hat{\mathbf{t}} - \mathbf{y}_j\|^2}. \tag{20}$$

Inlier Ratio refers to the percentage of estimated correspondences whose distance is less than a pre-defined threshold $\tau_1 = 0.1$m. Given point clouds $\mathbf{X}, \mathbf{Y}$ with an estimated correspondence set $\hat{\mathbf{C}} = \{(\mathbf{x}_i, \mathbf{y}_j) : \mathbf{x}_i \in \mathbf{X}, \mathbf{y}_j \in \mathbf{Y}\}$, ground-truth rotation $\mathbf{R} \in SO(3)$, and ground-truth translation $\mathbf{t} \in \mathbb{R}^3$, the IR of $\mathbf{X}, \mathbf{Y}$ is formulated as

$$\text{IR}(\mathbf{X}, \mathbf{Y}) = \frac{1}{|\hat{\mathbf{C}}|} \sum_{(\mathbf{x}_i, \mathbf{y}_j) \in \hat{\mathbf{C}}} \mathbb{1}\left(\|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_j\| < \tau_1\right). \tag{21}$$

Feature Matching Recall refers to the fraction of point cloud pairs with an IR$> \tau_2 = 5\%$.

We adopt RR for evaluation as it directly measures the performance on the target task of point cloud registration, while FMR and IR are not suitable to quantize our matching performance for three reasons: (1) our method detects only sparse keypoints for fine matching, hence it is not fair to be compared with other methods based on dense descriptors or correspondences when using the same number of sampled points for evaluation; (2) our method directly supervises the differential pose estimator to learn virtual correspondences with confidence scores instead of explicit feature matching and outlier rejection, hence our method inevitably retains many outliers which are acceptable for registration but not likely to maintain advanced matching performance; (3) feature matching recall and inlier ratio only reflect the matching performance which is not decisive for either accuracy (relative pose errors) or robustness (registration recall) especially for RANSAC-free methods [12, 13, 28, 49].

Table 8: Inlier ratios and feature matching recalls on indoor datasets 3DMatch.

	Inlier Ratio (%)										
	Benchmark	3DMatch					3DLoMatch				
	Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
descriptor-based	PerfectMatch [7]	36.0	32.5	26.4	21.5	16.4	11.4	10.1	8.0	6.4	4.8
	FCGF [8]	56.8	54.1	48.7	42.5	34.1	21.4	20.0	17.2	14.8	11.6
	D3Feat [9]	39.0	38.8	40.4	41.5	41.8	13.2	13.1	14.0	14.6	15.0
	SpinNet [19]	47.5	44.7	39.4	33.9	27.6	20.5	19.0	16.3	13.8	11.1
	YOHO [50]	64.4	60.7	55.7	46.4	41.2	25.9	23.3	22.6	18.2	15.0
	Predator [20]	58.0	58.4	57.1	54.1	49.3	26.7	28.1	28.3	27.5	25.8
correspondence-based	CoFiNet [11]	49.8	51.2	51.9	52.2	52.2	24.4	25.9	26.7	26.8	26.9
	GeoTransformer [12]	71.9	75.2	76.0	82.2	85.1	43.5	45.3	46.2	52.9	57.7
	OIF-Net [13]	62.3	65.2	66.8	67.1	67.5	27.5	30.0	31.2	32.6	33.1
	RoITr [28]	**82.6**	**82.8**	83.0	83.0	83.0	**54.3**	**54.6**	55.1	55.2	55.3
	PEAL [29]	74.8	81.3	86.0	87.9	89.2	49.1	54.1	60.5	63.6	65.0
	SIRA-PCR [49]	70.8	78.3	83.7	85.9	87.4	43.3	49.0	55.9	58.8	60.7
	DiffusionPCR [30]	75.0	81.6	86.3	88.2	89.4	49.7	55.4	61.8	64.5	66.2
	CAST	-	-	**91.2**	**91.5**	**93.1**	-	-	**66.3**	**66.3**	**66.5**
	Feature Matching Recall (%)										
	Benchmark	3DMatch					3DLoMatch				
	Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
descriptor-based	PerfectMatch[7]	95.0	94.3	92.9	90.1	82.9	63.6	61.7	53.6	45.2	34.2
	FCGF[8]	97.4	97.3	97.0	96.7	96.6	76.6	75.4	74.2	71.7	67.3
	D3Feat[9]	95.6	95.4	94.5	94.1	93.1	67.3	66.7	67.0	66.7	66.5
	SpinNet[19]	97.6	97.2	96.8	95.5	94.3	75.3	74.9	72.5	70.0	63.6
	YOHO[50]	98.2	97.6	97.5	97.7	96.0	79.4	78.1	76.3	73.8	69.1
	Predator[20]	96.6	96.6	96.5	96.3	96.5	78.6	77.4	76.3	75.7	75.3
correspondence-based	CoFiNet[11]	98.1	98.3	98.1	98.2	98.3	83.1	83.5	83.3	83.1	82.6
	GeoTransformer[12]	97.9	97.9	97.9	97.9	97.6	88.3	88.6	88.8	88.6	88.3
	OIF-Net [13]	98.1	98.1	97.9	98.4	98.4	84.6	85.2	85.5	86.6	87.0
	RoITr [28]	98.0	98.0	97.9	98.0	97.9	**89.6**	**89.6**	**89.5**	**89.4**	**89.3**
	PEAL [29]	**98.5**	**98.6**	**98.6**	**98.7**	**98.7**	89.1	89.2	89.0	89.0	88.8
	SIRA-PCR [49]	98.2	98.4	98.4	98.5	98.5	88.8	89.0	88.9	88.6	87.7
	DiffusionPCR [30]	98.3	98.3	98.3	98.3	98.3	86.3	85.9	86.0	86.1	85.9
	CAST	-	98.3	98.3	98.4	98.3	-	83.1	83.6	85.5	84.7

Feature Matching Performance. Nevertheless, this appendix presents IR and FMR in Table 8 to demonstrate the feature matching performance of our method. For IR evaluation, we scale the inlier confidences predicted by compatibility graph embedding to $[0, 1]$ and discard the correspondences with confidences < 0.1. As the number of keypoint correspondences after filtering is always less than 1000, we only report the IR regarding $\leq$1000 correspondences. Even without filtering, the number of keypoint correspondences is always less than 2500, hence we only report the FMR regarding $\leq$2500 correspondences. Enjoying the merits of our coarse matching, keypoint detection, and compatibility graph embedding, our method achieves the highest inlier ratio compared to all sorts of baselines. As for FMR, our method performs on par with DiffusionPCR [30] on 3DMatch and better than CoFiNet [11] on 3DLoMatch. However, our method performs worse than other coarse-to-fine methods [12, 13, 28, 29, 49, 30] on 3DLoMatch, since it remains challenging to extract enough keypoint correspondences in extremely low overlapping cases due to sparsity.

Indoor Registration Performance. We demonstrate the accuracy of CAST for indoor RGB-D point cloud registration by comparing it with various point cloud registration methods [15, 58, 59, 31, 32, 43, 60] in Table 9. All of the registration methods leverage the prevalent FCGF descriptor [8], and FastMAC [60] uses a sampling ratio of 50%. For a fair comparison, we follow the evaluation strategy of MAC [43] to re-compute the registration recall of our method, which is formulated as the fraction of point cloud pairs with RTE$<$30cm and RRE$<15°$. Our method achieves the highest registration recall and the lowest registration errors, suggesting its robustness and accuracy.

Table 9: Registration results on indoor RGBD point cloud datasets.

Methods	3DMatch			3DLoMatch		
	RR (%)	RTE (cm)	RRE (°)	RR (%)	RTE (cm)	RRE (°)
RANSAC-1M [15]	88.42	9.42	3.05	9.77	14.87	7.01
RANSAC-4M [15]	91.44	8.38	2.69	10.44	15.14	6.91
TEASER++ [58]	85.77	8.66	2.73	46.76	12.89	4.12
SC^2-PCR [59]	93.16	6.51	2.09	58.73	10.44	3.80
DGR [31]	88.85	7.02	2.28	43.80	10.82	4.17
PointDSC [32]	91.87	6.54	2.10	56.20	10.48	3.87
MAC [43]	93.72	6.54	2.02	59.85	9.75	3.50
FastMAC [60]	92.67	6.47	2.00	58.23	10.81	3.80
CAST	**96.48**	**5.64**	**1.71**	**76.13**	**8.47**	**2.75**

Table 10: Empirical standard deviations of the evaluation metrics of CAST in repeated experiments.

Dataset	3DMatch					KITTI			nuScenes		
Metrics	RR	IR	FMR	PIR	PMR	RR	RTE	RRE	RR	RTE	RRE
STD	0.4%	0.4%	0.5%	0.3%	0.2%	0.0%	0.1cm	0.01°	0.0%	0.1cm	0.01°

Generalization Studies. To extensively evaluate the generalizability of the proposed CAST in unseen domains, we conduct a generalization experiment from the outdoor dataset KITTI [39] to another outdoor dataset ETH [61]. Note that the KITTI and ETH datasets use Velodyne-64 3D LiDAR and Hokuyo 2D LiDAR, respectively, leading to very different appearances and distributions of point clouds. Hence, our generalization study is practical in applications and solid to demonstrate the generalizability of different methods. For fairness, all methods adopt 30cm for voxel down-sampling, and all methods involving RANSAC set the maximum iterations to be 50000 and the confidence to be 0.999 as the convergence criteria. To enhance the robustness, our method is combined with RANSAC estimating an initial pose from 250 coarse correspondences to reject the outliers during fine matching, and utilizes the global registration in GeoTransformer [12] to refine the pose estimate.

We present the translation errors, rotation errors, and the registration recalls in Table 11. Our method achieves satisfying accuracy and robustness, showcasing better generalizability than the coarse-to-fine baseline GeoTransformer [12] and other point-wise descriptors [8, 1, 20]. We also compare the our learnable compatibility graph embedding (CGE) with spectral matching (SM) [62] for outlier rejection in our method. With RANSAC filtering out severe outliers in advance, spectral matching can lead to better performance than learning-based CGE in unseen domains. Notably, all point-wise methods including CAST exhibit lower registration recalls in generalization studies than patch-wise local descriptor SpinNet [19] and BUFFER [48] incorporating patch-wise and point-wise features. This is mainly because they adopt a feature pyramid network architecture to learn features with abundant global context, which is detrimental for generalization [19]. Furthermore, we conduct an unsupervised domain adaptation (UDA) experiment for CAST, which tunes the network by learning to align a point cloud to itself after random rotation and cropping. The results indicate that our model can easily adapt to an unseen domain and achieve robust and accurate performance after one epoch's unsupervised tuning (only 20 minutes on an NVIDIA RTX3090 GPU).

Experiment Statistical Significance. Finally, Table 10 reports the standard deviations (1-sigma) of our evaluation metrics, which are assumed to be Gaussian distributed. Despite the randomness from voxel down-sampling and RANSAC, the performance of our method remains stable. Notably, the runtime of some methods such as [8, 19, 50] in Table 3 are quite different from results in [48], since we report the average runtime including data preprocessing, feature extraction, feature matching, and pose estimation, while the source codes of these methods save some intermediate results such as descriptors to avoid repeated calculation of the same point cloud in different pairs, which leads to unfair runtime comparison.

Table 11: Results of generalization from KITTI to ETH.

Methods	RTE (cm)	RRE (°)	RR (%)
FCGF [8]	9.08	0.94	45.86
Predator [20]	11.72	1.38	65.64
SpinNet [19]	6.05	0.98	99.44
TCKDD [1]	9.61	0.88	92.43
GeoTransformer [12]	5.97	0.73	91.87
BUFFER [48]	6.02	0.71	**100.00**
CAST (CGE)	6.85	0.65	97.76
CAST (SM)	6.66	0.61	98.04
CAST + UDA	**5.25**	**0.56**	99.58

A.4 Qualitative Results

Figure 3, Figure 7, and Figure 8 provide qualitative results about the registration performance on outdoor datasets KITTI [39], nuScenes [40], and the indoor dataset 3DMatch [16], respectively.

A.5 Limitation

The main limitation of the proposed CAST is the sub-optimal performance in low overlapping scenarios such as the 3DLoMatch benchmark compared to *state-of-the-art* methods, which may be ascribed to two aspects. (1) There is no effective outlier rejection for coarse matching as it is difficult to search inliers from patch correspondences based on geometry consistency due to low resolution. (2) Due to the sparsity and non-uniformity of keypoints, the inlier ratio of keypoint correspondences still falls short of what is required for robust pose estimation without a hypothesis-and-selection pipeline. Nevertheless, this RANSAC-free lightweight fine matching pipeline can achieve satisfying performance in outdoor scenarios. Considering the superior PIR and PMR of our coarse matching, we may directly exploit dense feature matching to enhance the robustness in low overlapping point cloud registration scenarios as a future work.

A.6 Broader Impacts

We present a novel consistency-aware spot-guided Transformer based on sparse attention to extract consistent coarse correspondences from point clouds. In addition, we propose a lightweight fine matching module for versatile and hierarchical point cloud registration, benefiting from the efficiency of sparse keypoint matching and the accuracy of dense registration. Different from existing methods, our fine matching is based on flexible local attention instead of optimal transport heavily relying on patch-to-patch correspondences, thus allowing independent deployment without coarse matching. Besides, the sparsity of keypoints ensures the efficiency of spatial consistency filtering.

Enjoying these merits, this work not only achieves superior accuracy, efficiency, and robustness in point cloud registration, but also paves the way to various large-scale real-time applications, such as SfM, SLAM, autonomous driving, or any other where point cloud registration plays a role. For examples, the reconstruction of indoor scenes and objects from unlabeled 3D scans could benefit from our work, which can precisely recover the rigid transformation between different scans. Additionally, our fine matching may independently construct a real-time LiDAR-based or RGBD camera-based odometry system for SLAM or SfM, as it is capable of efficient and reliable local data association and accurate pose estimation between two large-scale point clouds with a strong pose prior, while our coarse matching could be utilized in place recognition and global re-localization in SLAM.

As our work aims at tackling a fundamental problem in 3D computer vision, we do not anticipate a direct negative impact. Potential negative outcomes might occur in real applications where our method is involved.

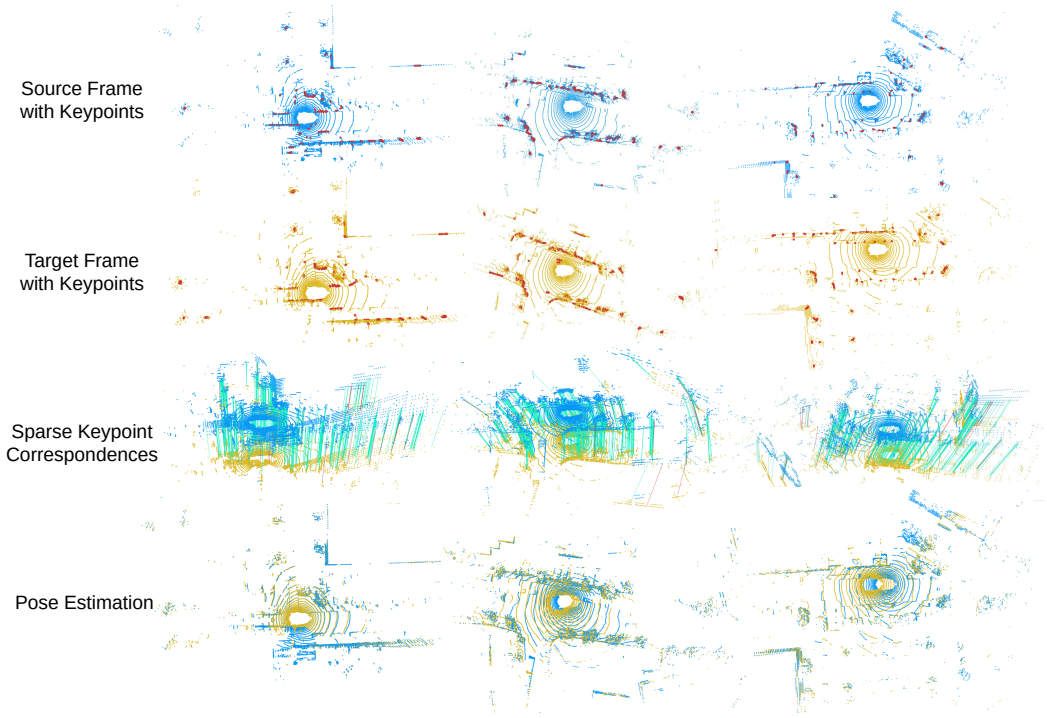

Figure 7: Qualitative registration results on nuScenes dataset. We show three examples in three columns to demonstrate the effectness of CAST in keypoint extraction, matching, and pose estimation.

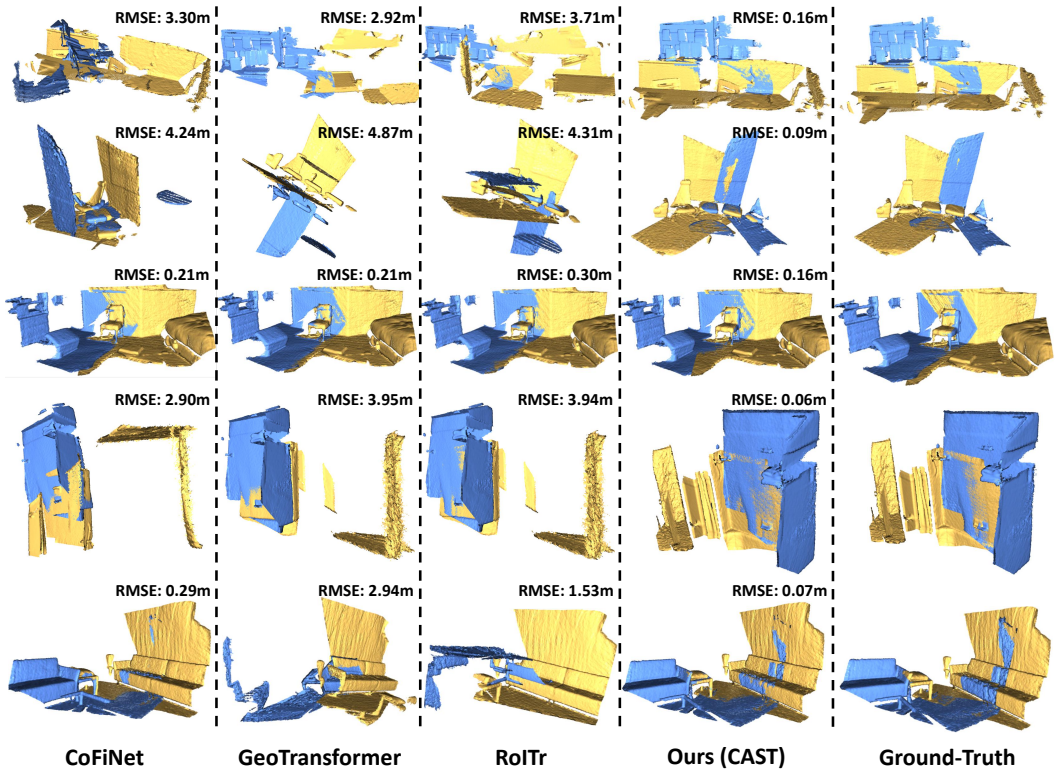

Figure 8: Qualitative registration results of CoFiNet [11], GeoTransformer [12], RoITr [28], and CAST compared with the ground truth alignment on 3DMatch dataset. We present five examples in five rows, which demonstrate the robustness and accuracy of our method.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction clearly state the claims made, including the motivations, contributions, and the performance of our approach.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have detailed the limitations of our work in Sec. A.5 of the appendix, and we also point out the scope of our claims made, including the benchmarks and sensors (Sec. A.2) and the computational efficiency (Sec. 4), *etc*.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include any theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: This paper fully discloses all the information needed to reproduce the main experimental results, including the detailed system architecture in Sec. 3 and some modular architectures along with the hyper-parameters in Sec. A.1, the training and evaluation settings in Sec. 4, and datasets with metrics in Sec. A.2 and Sec. A.3, respectively.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have submitted the data and codes in the supplementary material with detailed instructions on data access and preparation as well as guidelines to reproduce all experimental results. The paper will provide public access to the codes upon acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper fully specifies all the training and evaluation settings, including the optimizer and the learning rate scheduler in Sec. 4, hyper-parameters in Sec. A, data splits and preparation in Sec. A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In this paper, all metrics are assumed to be normally distributed, whose empirical standard deviations (1-sigma) in repeated experiments are reported in Table 10 to demonstrate the experiment statistical significance. The randomness mainly comes from voxel down-sampling and RANSAC.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Sec. 4, we have detailed the type of CPU and GPU of our device, and the runtime of the proposed method and nearly all baselines on our computer.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader impacts are discussed in Sec. A.6 of the appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: This paper involves existing benchmarks and utilizes existing methods for evaluation, whose papers are properly cited with licenses. The related URL are included in our codes.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: This paper provides details of our model in both main part (Sec. 3) and the appendix (Sec. A.1). Our source codes have been submitted in the supplementary material while no new datasets are proposed.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.